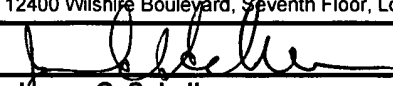
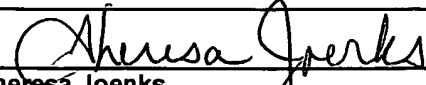


TRANSMITTAL FORM	Application Number	10/581,154
	Filing Date	31 May 2006
	First Named Inventor	Jiangang Zhuang
	Art Unit	
	Examiner Name	
(to be used for all correspondence after initial filing)		
Total Number of Pages in This Submission	36	Attorney Docket Number 42P23149

ENCLOSURES (Check all that apply)		
<input type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment/Reply <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input checked="" type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Reply to Missing Parts/ Incomplete Application <input type="checkbox"/> Reply to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____ <input type="checkbox"/> Landscape Table on CD	<input type="checkbox"/> After Allowance Communication to TC <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input checked="" type="checkbox"/> Other Enclosure(s) (please identify below): Return Receipt Postcard
Remarks		
SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT		
Firm Name	BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP 12400 Wilshire Boulevard, Seventh Floor, Los Angeles, CA 90025-1030	
Signature		
Printed name	James C. Scheller	
Date	7/6/2006	Reg. No. 31,195

CERTIFICATE OF TRANSMISSION/MAILING		
I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop PCT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.		
Signature		
Typed or printed name	Theresa Joenks	Date 7/6/06

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

中华人民共和国国家知识产权局
STATE INTELLECTUAL PROPERTY OFFICE
OF THE PEOPLE'S REPUBLIC OF CHINA



证 明 CERTIFICATE

本证明之附件是向中国专利局作为受理局提交的下列国际申请副本
THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY OF THE BELOW
IDENTIFIED INTERNATIONAL APPLICATION THAT WAS FILED WITH THE
CHINESE PATENT OFFICE AS RECEIVING OFFICE

国际申请号: PCT/CN2006/000163
INTERNATIONAL APPLICATION NUMBER
国际申请日: 26.1 月 2006 (26.01.2006)
INTERNATIONAL FILING DATE
发明名称: OPTIMIZING MEMORY ACCESSES FOR NETWORK
TITLE OF INVENTION APPLICATIONS USING INDEXED

CERTIFIED COPY OF
PRIORITY DOCUMENT

中华人民共和国国家知识产权局局长
COMMISSIONER OF THE STATE INTELLECTUAL PROPERTY
OFFICE OF THE PEOPLE'S REPUBLIC OF CHINA

二零零六年六月十五日

JUNE 15, 2006

PCT

REQUEST

The undersigned requests that the present international application be processed according to the Patent Cooperation Treaty.

For receiving Office use only

PCT/CN 2006 / 0 0 0 1 6 3
International Application No.
26 · 1月 2006 (26 · 01 · 2006)
International Filing Date
RO/CN 中华人民共和国国家知识产权局 PCT International Application
Name of receiving Office and "PCT International Application"
Applicant's or agent's file reference (if desired) (12 characters maximum) I06BJ1005

Box No. I TITLE OF INVENTION	
OPTIMIZING MEMORY ACCESSSES FOR NETWORK APPLICATIONS USING INDEXED REGISTER FILES	
Box No. II APPLICANT <input type="checkbox"/> This person is also inventor	
Name and address: (Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)	Telephone No.
INTEL CORPORATION 2200 Mission College Boulevard Santa Clara, California 95052 United States of America	Facsimile No.
	Teleprinter No.
	Applicant's registration No. with the Office
State (that is, country) of nationality: US	State (that is, country) of residence: US
This person is applicant for the purposes of: <input type="checkbox"/> all designated States <input checked="" type="checkbox"/> all designated States except the United States of America <input type="checkbox"/> the United States of America only <input type="checkbox"/> the States indicated in the Supplemental Box	
Box No. III FURTHER APPLICANT(S) AND/OR (FURTHER) INVENTOR(S)	
Name and address: (Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)	This person is:
ZHUANG, Jiangang No. 39, Beishan No. 1 Road, Ninghua County, Fujian 365400, P. R. China	<input type="checkbox"/> applicant only
	<input checked="" type="checkbox"/> applicant and inventor
	<input type="checkbox"/> inventor only (If this check-box is marked, do not fill in below.)
Applicant's registration No. with the Office	
State (that is, country) of nationality: CN	State (that is, country) of residence: CN
This person is applicant for the purposes of: <input type="checkbox"/> all designated States <input type="checkbox"/> all designated States except the United States of America <input checked="" type="checkbox"/> the United States of America only <input type="checkbox"/> the States indicated in the Supplemental Box	
<input checked="" type="checkbox"/> Further applicants and/or (further) inventors are indicated on a continuation sheet.	
Box No. IV AGENT OR COMMON REPRESENTATIVE; OR ADDRESS FOR CORRESPONDENCE	
The person identified below is hereby/has been appointed to act on behalf of the applicant(s) before the competent International Authorities as:	<input checked="" type="checkbox"/> agent <input type="checkbox"/> common representative
Name and address: (Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)	Telephone No.
IntellecPro China Limited Suites 902-908, Ping'an Mansion, 23 Jinrong Dajie, Xicheng District, Beijing 100032, P. R. China	86-10-66215588
	Facsimile No.
	86-10-66210771
Teleprinter No.	
Agent's registration No. with the Office	
11015	
<input type="checkbox"/> Address for correspondence: Mark this check-box where no agent or common representative is/has been appointed and the space above is used instead to indicate a special address to which correspondence should be sent.	

Continuation of Box No. III FURTHER APPLICANT(S) AND/OR (FURTHER) INVENTOR(S)

If none of the following sub-boxes is used, this sheet should not be included in the request.

Name and address: (Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)

DAI, Jinquan
Room 309, Block 2, Ren Le Xin Cun, Songjiang District,
Shanghai 201600, P. R. China

This person is:

- ☐ applicant only
☒ applicant and inventor
☐ inventor only (If this check-box is marked, do not fill in below.)

Applicant's registration No. with the Office

State (that is, country) of nationality:

CN

State (that is, country) of residence:

CN

This person is applicant for the purposes of:

- ☐ all designated States ☐ all designated States except the United States of America ☒ the United States of America only ☐ the States indicated in the Supplemental Box

Name and address: (Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)

LI, Long
Room 202, Building 34, No. 123 Fuquan Road, Shanghai
200335, P. R. China

This person is:

- ☐ applicant only
☒ applicant and inventor
☐ inventor only (If this check-box is marked, do not fill in below.)

Applicant's registration No. with the Office

State (that is, country) of nationality:

CN

State (that is, country) of residence:

CN

This person is applicant for the purposes of:

- ☐ all designated States ☐ all designated States except the United States of America ☒ the United States of America only ☐ the States indicated in the Supplemental Box

Name and address: (Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)

This person is:

- ☐ applicant only
☐ applicant and inventor
☐ inventor only (If this check-box is marked, do not fill in below.)

Applicant's registration No. with the Office

State (that is, country) of nationality:

State (that is, country) of residence:

This person is applicant for the purposes of:

- ☐ all designated States ☐ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

Name and address: (Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)

This person is:

- ☐ applicant only
☐ applicant and inventor
☐ inventor only (If this check-box is marked, do not fill in below.)

Applicant's registration No. with the Office

State (that is, country) of nationality:

State (that is, country) of residence:

This person is applicant for the purposes of:

- ☐ all designated States ☐ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

☐ Further applicants and/or (further) inventors are indicated on another continuation sheet.

Box No. V DESIGNATIONS

The filing of this request constitutes under Rule 4.9(a), the designation of all Contracting States bound by the PCT on the international filing date, for the grant of every kind of protection available and, where applicable, for the grant of both regional and national patents.

However,

- ☐ DE Germany is **not designated** for any kind of national protection
- ☐ KR Republic of Korea is **not designated** for any kind of national protection
- ☐ RU Russian Federation is **not designated** for any kind of national protection

(The check-boxes above may be used to exclude (irrevocably) the designations concerned in order to avoid the ceasing of the effect, under the national law, of an earlier national application from which priority is claimed. See the Notes to Box No. V as to the consequences of such national law provisions in these and certain other States.)

Box No. VI PRIORITY CLAIM

The priority of the following earlier application(s) is hereby claimed:

Filing date of earlier application (day/month/year)	Number of earlier application	Where earlier application is:		
		national application: country or Member of WTO	regional application:* regional Office	international application: receiving Office
item (1)				
item (2)				
item (3)				

- ☐ Further priority claims are indicated in the Supplemental Box.

The receiving Office is requested to prepare and transmit to the International Bureau a certified copy of the earlier application(s) (only if the earlier application was filed with the Office which for the purposes of this international application is the receiving Office) identified above as:

- ☐ all items ☐ item (1) ☐ item (2) ☐ item (3) ☐ other, see Supplemental Box

* Where the earlier application is an ARIPO application, indicate at least one country party to the Paris Convention for the Protection of Industrial Property or one Member of the World Trade Organization for which that earlier application was filed (Rule 4.10(b)(ii)).

Box No. VII INTERNATIONAL SEARCHING AUTHORITY

Choice of International Searching Authority (ISA) (if two or more International Searching Authorities are competent to carry out the international search, indicate the Authority chosen; the two-letter code may be used):

ISA / CN

Request to use results of earlier search; reference to that search (if an earlier search has been carried out by or requested from the International Searching Authority):

Date (day/month/year) Number Country (or regional Office)

Box No. VIII DECLARATIONS

The following declarations are contained in Boxes Nos. VIII (i) to (v) (mark the applicable check-boxes below and indicate in the right column the number of each type of declaration):

Number of
declarations

- | | | |
|---|--|---|
| <input type="checkbox"/> Box No. VIII (i) | Declaration as to the identity of the inventor | : |
| <input type="checkbox"/> Box No. VIII (ii) | Declaration as to the applicant's entitlement, as at the international filing date, to apply for and be granted a patent | : |
| <input type="checkbox"/> Box No. VIII (iii) | Declaration as to the applicant's entitlement, as at the international filing date, to claim the priority of the earlier application | : |
| <input type="checkbox"/> Box No. VIII (iv) | Declaration of inventorship (only for the purposes of the designation of the United States of America) | : |
| <input type="checkbox"/> Box No. VIII (v) | Declaration as to non-prejudicial disclosures or exceptions to lack of novelty | : |

Box No. IX CHECK LIST; LANGUAGE OF FILING

This international application contains:

(a) on paper, the following number of sheets:

request (including declaration sheets) : 4
 description (excluding sequence listing and/or tables related thereto) : 13
 claims : 4
 abstract : 1
 drawings : 7

Sub-total number of sheets : 29

sequence listing :
 tables related thereto :

(for both, actual number of sheets if filed on paper, whether or not also filed in electronic form; see (c) below)

Total number of sheets : 29

(b) ☐ only in electronic form (Section 801(a)(i))

(i) ☐ sequence listing
 (ii) ☐ tables related thereto

(c) ☐ also in electronic form (Section 801(a)(ii))

(i) ☐ sequence listing
 (ii) ☐ tables related thereto

Type and number of carriers (diskette, CD-ROM, CD-R or other) on which are contained the

☐ sequence listing :
☐ tables related thereto :

(additional copies to be indicated under items 9(ii) and/or 10(ii), in right column)

This international application is accompanied by the following item(s) (mark the applicable check-boxes below and indicate in right column the number of each item):

1. ☒ fee calculation sheet : 1
 2. ☐ original separate power of attorney :
 3. ☐ original general power of attorney :
 4. ☐ copy of general power of attorney; reference number, if any: :
 5. ☐ statement explaining lack of signature :
 6. ☐ priority document(s) identified in Box No. VI as item(s): :
 7. ☐ translation of international application into (language): :
 8. ☐ separate indications concerning deposited microorganism or other biological material :
 9. ☐ sequence listing in electronic form (indicate type and number of carriers)
 (i) ☐ copy submitted for the purposes of international search under Rule 13ter only (and not as part of the international application) :
 (ii) ☐ (only where check-box (b)(i) or (c)(i) is marked in left column) additional copies including, where applicable, the copy for the purposes of international search under Rule 13ter :
 (iii) ☐ together with relevant statement as to the identity of the copy or copies with the sequence listing mentioned in left column :
 10. ☐ tables in electronic form related to sequence listing (indicate type and number of carriers)
 (i) ☐ copy submitted for the purposes of international search under Section 802(b-quater) only (and not as part of the international application) :
 (ii) ☐ (only where check-box (b)(ii) or (c)(ii) is marked in left column) additional copies including, where applicable, the copy for the purposes of international search under Section 802(b-quater) :
 (iii) ☐ together with relevant statement as to the identity of the copy or copies with the tables mentioned in left column :
 11. ☐ other (specify): :

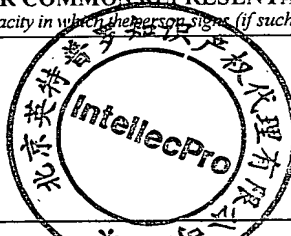
Figure of the drawings which should accompany the abstract:

Language of filing of the international application:

English

Box No. X SIGNATURE OF APPLICANT, AGENT OR COMMON REPRESENTATIVE

Next to each signature, indicate the name of the person signing and the capacity in which the person signs (if such capacity is not obvious from reading the request).



IntellecPro China Limited

For receiving Office use only

1. Date of actual receipt of the purported international application: 26. 1月 2006 (26. 01. 2006)	2. Drawings: <input type="checkbox"/> received: <input type="checkbox"/> not received:
3. Corrected date of actual receipt due to later but timely received papers or drawings completing the purported international application:	
4. Date of timely receipt of the required corrections under PCT Article 11(2):	
5. International Searching Authority (if two or more are competent): ISA /	
6. <input type="checkbox"/> Transmittal of search copy delayed until search fee is paid	

For International Bureau use only

Date of receipt of the record copy by the International Bureau:

This sheet is not part of and does not count as a sheet of the international application.

PCT

FEE CALCULATION SHEET

Annex to the Request

For receiving Office use only

PCT/CN 2006 / 0 0 0 1 6 3
International Application No.

26 · 1月 2006 (26 · 01 · 2006)
Date stamp of the receiving Office

Applicant's or agent's
file reference

I06BJ1005

Applicant

INTEL CORPORATION, et al.

CALCULATION OF PRESCRIBED FEES

1. TRANSMITTAL FEE

CNY 500 T

2. SEARCH FEE

CNY 2100 S

International search to be carried out by CN

(If two or more International Searching Authorities are competent to carry out the international search, indicate the name of the Authority which is chosen to carry out the international search.)

3. INTERNATIONAL FILING FEE

Where items (b) and/or (c) of Box No. IX apply, enter Sub-total number of sheets } 29
Where items (b) and (c) of Box No. IX do not apply, enter Total number of sheets }

i1 first 30 sheets CHF 1400 i1

i2 number of sheets in excess of 30 x fee per sheet = i2

i3 additional component (only if a sequence listing and/or tables related thereto are filed in electronic form under Section 801(a)(i), or both in that form and on paper, under Section 801(a)(ii)):

400 x fee per sheet = i3

Add amounts entered at i1, i2 and i3 and enter total at I

CHF 1400 I

(Applicants from certain States are entitled to a reduction of 75% of the international filing fee. Where the applicant is (or all applicants are) so entitled, the total to be entered at I is 25% of the international filing fee.)

4. FEE FOR PRIORITY DOCUMENT (if applicable)

5. TOTAL FEES PAYABLE

Add amounts entered at T, S, I and P, and enter total in the TOTAL box

CNY 2600 CHF 1400
TOTAL

MODE OF PAYMENT (Not all modes of payment may be available at all receiving Offices)

- ☒ authorization to charge deposit account (see below) ☐ postal money order ☐ cash ☐ coupons
☐ cheque ☐ bank draft ☐ revenue stamps ☐ other (specify):

AUTHORIZATION TO CHARGE (OR CREDIT) DEPOSIT ACCOUNT

(This mode of payment may not be available at all receiving Offices)

- ☒ Authorization to charge the total fees indicated above.
☐ (This check-box may be marked only if the conditions for deposit accounts of the receiving Office so permit) Authorization to charge any deficiency or credit any overpayment in the total fees indicated above.
☐ Authorization to charge the fee for priority document.

Receiving Office: RO/ CN

Deposit Account No.:

Date:

Name:

Signature:



OPTIMIZING MEMORY ACCESSES FOR NETWORK APPLICATIONS USING INDEXED REGISTER FILES

Background

Field

[0001] The embodiments relate to high-speed network devices, and more particularly to optimizing memory access for high-speed network devices.

Description of the Related Art

[0002] Synchronous optical network (SONET) is a standard for optical telecommunications transport formulated by the Exchange Carriers Standards Association (ECSA) for the American National Standards Institute (ANSI), which sets industry standards in the U.S. for telecommunications and other industries. Network processors (NP) are emerging as a core element of network devices, such as high-speed communication routers. NPs are designed specifically for network processing applications.

[0003] The unique challenge of network processing is to guarantee and sustain throughput for the worst-case traffic. For instance, the case of the optical level OC-192 (10 Gigabits/sec) POS (Packet over SONET) packet processing presents significant processing and throughput challenges. It requires a throughput of 28 million packets per second or service time of 4.57 microseconds per packet for processing in the worst case. The latency for a single external memory access is much larger than the worst-case service time.

[0004] Therefore, modern network processors usually have a highly

parallel architecture with non-uniform memory hierarchy. Network processors can consist of multiple microengines (*MEs*, or programmable processors with packet processing capability) running in parallel. Each *ME* has its own local memory (*LM*), for example registers.

[0005] Various constraints may be applied to accessing register files, which complicates the management of the register files. For example, a local memory in a NP can be addressed using a *BASE-OFFSET* word address. The *BASE* value is stored in a specific base-address register, and there is 3-cycle latency between writing the base-address register when its value changes.

[0006] The *OFFSET* is a constant from 0 to 15. The final address in the *BASE-OFFSET* mode, however, is computed using a logical *OR* operation (i.e., $BASE \mid OFFSET$). Therefore, to support C pointer arithmetic, e.g., *pointer + offset*, using the *BASE-OFFSET* mode of local memory where $BASE = pointer$ and $OFFSET = offset$, proper alignment of *BASE* has to be ensured such that the condition in **Fig. 1** holds. Otherwise, to access that address, the base-register has to be set to $pointer + offset$, and the *OFFSET* is set to 0. **Fig. 1** illustrates the alignment requirement of the *BASE-OFFSET* addressing mode of the local memory.



CONTENTS OF THE INVENTION

Problem to be solved

[0007] Current network processors (NP) have latency between writing the base-address register and when its value changes. Further latency is added when accessing external memory to the NP. Therefore, the problem is how to reduce latency with memory accesses.

Solutions

[0008] In order to improve performance for network applications, one embodiment includes an optimizing compiler to optimize and minimize external memory accesses using the local memory (i.e., indexed register files), and minimizes the initializations of the base-address register for efficient local memory accesses.

[0009] One embodiment migrates external memory objects (e.g., variables) to the local memory (i.e., indexed register files), and optimizes the accesses to the local memory by determining alignment of the migrated objects; and eliminating redundant initialization code of the objects.

[0010] The advantages of the embodied solutions is that objects that are accessed from external memory are now accessed through local memory to a network processor (e.g., indexed registers) and the latency from writing the base-address register when its value changes is reduced as redundant initializations are eliminated.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The embodiments are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:



- [0012] Fig. 1 illustrates the alignment requirement of the BASE-OFFSET addressing mode of the local memory;
- [0013] Fig. 2 illustrates a block diagram of an embodiment;
- [0014] Fig. 3 illustrates a block diagram of object migration;
- [0015] Fig. 4A-B illustrates original object and code sequences in external memory;
- [0016] Fig. 5A-B illustrates objects migrated to local memory and access code sequence without alignment adjustment;
- [0017] Fig. 6 illustrates a block diagram for determining alignment of migrated objects;
- [0018] Fig. 7 illustrates pseudo-code for determining and setting the minimum alignment needed for each object;
- [0019] Fig. 8A-B illustrates objects migrated to local memory and access code sequence with alignment adjusted;
- [0020] Fig. 9A-B illustrates objects migrated to local memory and access code sequence with alignment adjusted and redundant initializations eliminated;
- [0021] Fig. 10 illustrates an embodiment of a processing device;
- [0022] Fig. 11 illustrates an optimizer system for network processors;
and
- [0023] Fig. 12 illustrates a compiler of the embodiment illustrated in Fig. 10.

DETAILED DESCRIPTION

[0024] The embodiments discussed herein generally relate to optimization of local memory accessing and latency reduction for network processors. Referring to the figures, exemplary embodiments will now be described. The exemplary embodiments are provided to illustrate the embodiments and should not be construed as limiting the scope of the embodiments.

[0025] Fig. 2 illustrates a general block diagram of an embodiment for optimizing an executable. In this embodiment external memory accesses are optimized using indexed register files, by efficiently migrating external memory variables to the local memory (i.e., indexed register files), and minimizing the initializations of the base-address register of the local memory. In block 1 objects that change values (e.g., variables) are migrated from an external memory, such as a main memory, such as random-access memory (RAM), static random access memory (SRAM), dynamic random access memory (DRAM), synchronous DRAM (SDRAM), read-only memory (ROM), etc., of a network processor (NP) to local memory of the NP, such as indexed registers. In block 2, alignment of the migrated objects is determined. And, in block 3 redundant initialization code of the objects is eliminated. Fig. 3 illustrates block 1 in further detail.

[0026] Fig. 3 illustrates a block diagram of object migration. In general, the blocks in Fig. 3 cover determining whether each object of the plurality of objects are accessible from a plurality of processors; determining an equivalence set of aliased objects in the plurality of objects; determining objects of the plurality of objects eligible for migration; changing residence of the objects determined to be eligible for migration; and changing accesses of the objects having their residence changed.

[0027] As illustrated in Fig. 2, first variables are migrated from external

memory to local memory (i.e., indexed registers). In one embodiment, the eligible objects (i.e., variables) in external memory are migrated to the local memory. That is, the residences of those variables are changed to the local memory, and the accesses to those variables are changed correspondingly.

[0028] Since local memory resides in each NP and the local memory in one processor cannot be shared with another processor, variables that are accessed by multiple processors are not migrated to local memory. In block 1.1 it is determined whether a variable is accessed by multiple processors through escape analysis. In one embodiment, escape analysis determines whether an object (i.e., variable) is accessed by more than one processor. Consequently, variables in external memory can be migrated to indexed register files for fast accesses, no matter whether they are accessed using constant addresses or pointers (i.e., non-constant addresses).

[0029] In block 1.2 an equivalence set of aliased variables are computed through points-to analysis. That is, variables that could possibly be accessed by one instruction belong to the same equivalence set. If one variable in an equivalence set cannot be migrated to local memory, none of those variables in the same equivalence set can be migrated to local memory. In one embodiment the total size of variables should not exceed the available local memory size. With the above constraints and the equivalence set, variables that are eligible for migration are computed in block 1.3.

[0030] In block 1.4, the residence of eligible variables is changed from external memory to local memory. In block 1.5, accesses of those variables whose residence were changed is changed.

[0031] For example, suppose there are three variables A, B, C in an external memory (e.g. SRAM) whose original alignment and size are illustrated in Fig. 4A. The access order is illustrated in Fig. 4B. For the access order of A, B, and C as illustrated in Fig. 4B, several I/O operations to the external

memory are needed.

[0032] Fig. 5A-B illustrate data migrated to local memory and access code sequence without alignment adjustment. For ease of discussion, suppose A, B, and C satisfy the migration condition, after migrating these objects to local memory, accesses of these objects with local memory base address register initialization code inserted are illustrated in Fig. 5B. Without further optimization, none of the accesses can share the base address value because the base address and offset value do not satisfy the alignment requirement of the BASE-OFFSET mode.

[0033] Fig. 6 illustrates a block diagram of determining the alignment of migrated variables through a forward disjunctive dataflow analysis. In one embodiment, the alignments of the migrated objects are adjusted properly, such that the sharing of the base address register is maximized between the accesses to the local memory. In one embodiment, the minimum alignments required for objects in local memory to maximize sharing of base registers and to reduce padding between variables are determined. That is, the alignment of the object in local memory is determined such that, any smaller alignment causes less sharing of the base-address register, and any alignment larger than this value does not cause more sharing of the base-address address.

[0034] Block 2.1 uses a forward disjunctive dataflow analysis to compute the offset value pairs with a common base address. The dataflow analysis uses a simplified flow graph, i.e., those instructions that do not contain any accesses to migrated objects are purged off and each flow node consists of only one instruction.

[0035] In the simplified flow graph, flow nodes and instructions are the same. In one embodiment, it is assumed that each instruction contains, at most, one local memory access, and the address of the access is expressed in the form of *base address + constant offset*. The dataflow equations for each instruction i is shown below.

$$\begin{cases} GEN[i] = \{ L \mid L \text{ is the local memory base address used in instruction } i \} \\ KILL[i] = \{ L \mid L \text{ is the local memory base address not used in instruction } i \} \\ IN[i] = \bigcup_{p \in Pred(i)} OUT[p] \\ OUT[i] = GEN[i] \cup (IN[i] - KILL[i]) \end{cases}$$

[0036] The forward disjunctive dataflow analysis is iterated until both IN and OUT are converged. For the example of sequential accesses illustrated in Fig. 5B, the values of GEN and KILL for each local memory access are as follows:

$$\begin{array}{ll} GEN(1) = \{A[i][0]\} & KILL(1) = \{B[i][0], C[i][0]\} \\ GEN(2) = \{A[i][0]\} & KILL(2) = \{B[i][0], C[i][0]\} \\ GEN(3) = \{B[i][0]\} & KILL(3) = \{A[i][0], C[i][0]\} \\ GEN(4) = \{B[i][0]\} & KILL(4) = \{A[i][0], C[i][0]\} \\ GEN(5) = \{B[i][0]\} & KILL(5) = \{A[i][0], C[i][0]\} \\ GEN(6) = \{B[i][0]\} & KILL(6) = \{A[i][0], C[i][0]\} \\ GEN(7) = \{C[i][0]\} & KILL(7) = \{A[i][0], B[i][0]\} \\ GEN(8) = \{C[i][0]\} & KILL(8) = \{A[i][0], B[i][0]\} \end{array}$$

[0037] The final values of IN and OUT are as follows:

$$\begin{array}{ll} IN(1) = \{ \} & OUT(1) = \{A[i][0]\} \\ IN(2) = \{A[i][0]\} & OUT(2) = \{A[i][0]\} \\ IN(3) = \{A[i][0]\} & OUT(3) = \{B[i][0]\} \\ IN(4) = \{B[i][0]\} & OUT(4) = \{B[i][0]\} \\ IN(5) = \{B[i][0]\} & OUT(5) = \{B[i][0]\} \\ IN(6) = \{B[i][0]\} & OUT(6) = \{B[i][0]\} \\ IN(7) = \{B[i][0]\} & OUT(7) = \{C[i][0]\} \\ IN(8) = \{C[i][0]\} & OUT(8) = \{C[i][0]\} \end{array}$$

[0038] In one embodiment, each base address in $GEN[i] \cap IN[i]$ is used by two consecutive local memory accesses to the same object, with possibly different (constant) offsets. In one embodiment, if the base address and one of the constant offsets do not satisfy the requirement in Fig. 1, the sharing of the base address is not possible. In this case, one embodiment of an optimizer (e.g., compiler) can enlarge the alignment to the objects so that the base address and offset values can meet the alignment requirement.

[0039] In one embodiment, the pair of two different offset values (*offset value pair*) of the two consecutive local memory accesses that use the same base address can be computed during the dataflow iteration. That is, when calculating the IN set for flow node i , if $GEN[i] \cap IN[i]$ is found not to be empty, the different offset values of the current and previous local memory accesses (that use the same base address) are recorded as a pair of offset values (associated with the base address). In the above example, the list of offset value pairs associated with the base address is shown below.

```
A[i][0] -> {(0, 4)}
B[i][0] -> {(0, 4), (4, 8), (8, 12)}
C[i][0] -> {(0, 4)}
```

[0040] For each base address, assume VAR is a variable accessed by this base address and its size is $SIZE$; then the upper bound of the alignment to be attempted for VAR , or $MAX_ALIGN(VAR)$, can be determined as follows. Here the MAX_ALIGN is the width (in bytes) of the $OFFSET$ in the $BASE-OFFSET$ addressing mode (for instance, 64 for the local memory in a NP).



$$MAX_ALIGN(VAR) = \min \left(MAX_ALIGN, 2^{\lceil \log_2 SIZE \rceil} \right)$$

[0041] Fig. 7 illustrates pseudo code of block 2.2 that determines and sets the minimum alignment needed for each object (i.e., variable) in local memory. In one embodiment, with the pseudo-code in Fig. 7, the alignment of the objects can be properly set so that two consecutive accesses of the same variable can use the same base address value, thus reducing the number of base address register initialization instructions. According to the definition of the minimum alignment and the pseudo-code illustrated in Fig. 7, the minimum alignments of A, B, C are set to 8, 16 and 8 bytes, respectively. With the adjusted alignment, the data in local memory and access code sequence are illustrated in Fig. 8A-B.

[0042] The result of block 2.2 (illustrated in Fig. 6) could contain some redundant initializations of the base address register. In one embodiment, those redundant initializations are eliminated using any existing (partial) redundancy elimination algorithms. After the redundant initialization code elimination, the data in local memory and the access code sequence is illustrated in Fig. 9A-B, in which the initialization instructions are greatly reduced.

[0043] Fig. 10 illustrates a processing device 1000. Processing device 1000 includes a processing device (e.g., a NP) 1020 connected to external memory 1010 (e.g., SDRAM). Processing device 1000 further includes optimizer 1040 that is coupled to processor 1030. Further included are indexed registers 1050 coupled to processing device 1020. As illustrated, objects 1060 currently reside in indexed registers 1060 after being migrated from external memory 1010. Optimizer 1040 minimizes external memory 1010 accesses by

migrating objects 1060 (e.g., variables) in external memory to indexed registers 1050 (e.g., the local memory on processing device 1020). That is, changing the residence of those objects to indexed registers 1050 and the accesses to those objects. Consequently, external memory 1010 access latency is minimized for network applications. In one embodiment, processing device 1000 is a high-speed networking router. On another embodiment, multiple processing devices 1020 are included in processing device 1000.

[0044] Fig. 11 illustrates an optimizer system 1100. Optimizer system 1100 includes a processing device including processor 1110 coupled to memory 1120, display 1140 and optimizer 1130. In one embodiment, optimizer 1130 is a compiler. Display 1140 can be any known type of display device, such as liquid crystal display (LCD), cathode ray tube (CRT), flat screen technology, projection display, etc. Optimizer system further is coupled to processing device 1000 that does not include optimizer 1040. In one embodiment, optimizer system 1100 is removably coupled to processing device 1000. That is, optimizer system can be coupled to processing device 1000 with a cable, through a network, wireless connection, etc. In one embodiment, optimizer 1130 optimizes processing device 1000 by migrating objects from external memory 1010 to local memory (i.e., indexed registers) 1050 using escape analysis and points-to analysis in a straightforward way. Computation of the minimal alignments of objects (i.e., variables) in local memory 1050 to maximize sharing of base registers and reduce padding between objects through a forward disjunctive dataflow analysis which takes the code proximity into account. Optimizer 1130 further minimizes costly initialization operations through an existing redundancy elimination algorithm.

[0045] Fig. 12 illustrates components of optimizer 1130. In one embodiment optimizer 1130 includes first determiner 1210 that determines whether each object (e.g., variable) of a plurality of objects are accessible from more than one processor in a network device. Second determiner 1220



determines an equivalence set of aliased objects in the plurality of objects.

Third determiner 1230 determines objects of the plurality of objects eligible for migration. Migrator 1240 changes residence of the objects determined to be eligible for migration. Accessor 1250 changes accesses of the objects having their residence changed. Optimizer 1130 maximizes the sharing of the base-address registers and minimizes the padding between objects in indexed registers 1050 by properly adjusting the alignments of those objects. Together with redundancy elimination, optimizer 1130 minimizes the initializations of the base-address registers.

[0046] Embodiments of the present disclosure described herein may be implemented in circuitry, which includes hardwired circuitry, digital circuitry, analog circuitry, programmable circuitry, and so forth. These embodiments may also be implemented in computer programs. Such computer programs may be coded in a high level procedural or object oriented programming language. The program(s), however, can be implemented in assembly or machine language if desired. The language may be compiled or interpreted. Additionally, these techniques may be used in a wide variety of networking environments. Such computer programs may be stored on a storage media or device (e.g., hard disk drive, floppy disk drive, read only memory (ROM), CD-ROM device, flash memory device, digital versatile disk (DVD), or other storage device) readable by a general or special purpose programmable processing system, for configuring and operating the processing system when the storage media or device is read by the processing system to perform the procedures described herein. Embodiments of the disclosure may also be considered to be implemented as a machine-readable or machine recordable storage medium, configured for use with a processing system, where the storage medium so configured causes the processing system to operate in a specific and predefined manner to perform the functions described herein.

[0047] While certain exemplary embodiments have been described and

4D

shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not restrictive on the broad invention, and that this invention not be limited to the specific constructions and arrangements shown and described, since various other modifications may occur to those ordinarily skilled in the art.

[0048] Reference in the specification to "an embodiment," "one embodiment," "some embodiments," or "other embodiments" means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least some embodiments, but not necessarily all embodiments. The various appearances "an embodiment," "one embodiment," or "some embodiments" are not necessarily all referring to the same embodiments. If the specification states a component, feature, structure, or characteristic "may", "might", or "could" be included, that particular component, feature, structure, or characteristic is not required to be included. If the specification or claim refers to "a" or "an" element, that does not mean there is only one of the element. If the specification or claims refer to "an additional" element, that does not preclude there being more than one of the additional element.



CLAIMS

1. A method for optimizing an executable comprising:
migrating a plurality of objects from a first memory to a second memory;
determining alignment of the migrated plurality of objects; and
eliminating redundant initialization code of the plurality of objects.
2. The method for optimizing an executable of claim 1, wherein the plurality of objects are variables.
3. The method for optimizing an executable of claim 1, the migrating the plurality of objects further comprising:
determining whether each object of the plurality of objects are accessible from a plurality of processors;
determining an equivalence set of aliased objects in the plurality of objects;
determining objects of the plurality of objects eligible for migration;
changing residence of the objects determined to be eligible for migration;
and
changing accesses of the objects having their residence changed.
4. The method for optimizing an executable of claim 1, the determining alignment further comprising:
analyzing the migrated objects by forward disjunctive dataflow analysis;
determining a minimum alignment necessary for each migrated object;
and
setting the minimum alignment necessary for each migrated object.
5. The method for optimizing an executable of claim 1, wherein the first

4/5

memory is an external memory and the second memory comprises a plurality of indexed registers residing in a microengine.

6. A processing device comprising:

an optimizer to migrate a plurality of objects from an external memory of a network processing device to a plurality of registers coupled to a processor, the optimizer further to align and eliminate redundant initialization code of the plurality of objects.

7. The processing device of claim 6, wherein the plurality of registers are indexed.

8. The processing device of claim 6, wherein the plurality of objects are variables.

9. The processing device of claim 6, wherein the migrated plurality of objects are not shared by the processor and at least one other processor.

10. The processing device of claim 6, wherein the network processing device is a router.

11. An optimizer system for network processors comprising:

a processor,

a first memory coupled to the processor;

a display coupled to the processor; and

a compiler to migrate a plurality of objects from a second memory to a plurality of indexed registers in a network processor, the compiler further to align and eliminate redundant initialization code of the plurality of objects.

12. The optimizer system for network processors of claim 11, the compiler



including:

a first determiner to determine whether each object of the plurality of objects are accessible from a plurality of processors in a network device;

a second determiner to determine an equivalence set of aliased objects in the plurality of objects;

a third determiner to determine objects of the plurality of objects eligible for migration;

a migrator to change residence of the objects determined to be eligible for migration; and

an accessor to change accesses of the objects having their residence changed.

13. The optimizer system for network processors of claim 12, wherein the second memory is external to the plurality of processors.

14. The optimizer system for network processors of claim 11, wherein the plurality of objects are variables.

15. The optimizer system for network processors of claim 11, wherein the second memory is external to the plurality of indexed registers.

16. A machine-accessible medium containing instructions that, when executed, cause a machine to:

migrate a plurality of variables from a first memory to a plurality of indexed registers;

align the migrated plurality of variables; and

eliminate redundant initializations to a base address register.

17. The machine accessible medium of claim 16, further comprising instructions that, when executed, cause a machine to:



determine whether each variable of the plurality of variables are accessible from at least two network processors;

determine an equivalence set of aliased variables in the plurality of variables; and

change location of the variables that are determined to be eligible for migration.

18. The machine accessible medium of claim 16, further comprising instructions that, when executed, cause a machine to:

analyze the migrated variables by forward disjunctive dataflow analysis;
and

determine a minimum alignment necessary for each migrated variable.

19. The machine accessible medium of claim 16, further comprising instructions that, when executed, cause a machine to:

set the minimum alignment necessary for each migrated variable.

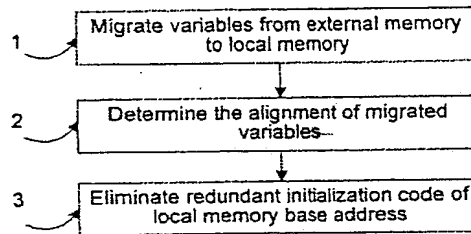
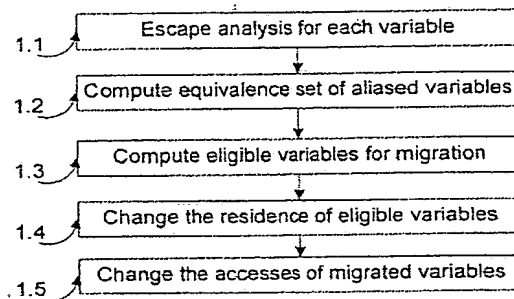
20. The machine accessible medium of claim 16, further comprising instructions that, when executed, cause a machine to:

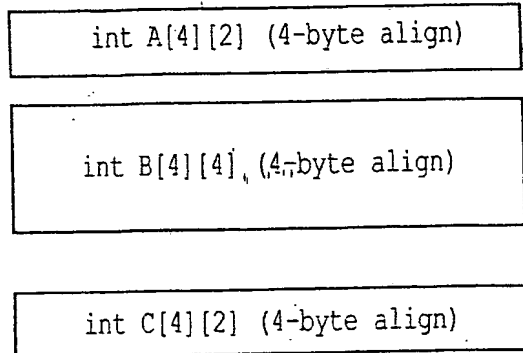
compile source code to migrate the plurality of variables from the first memory to the plurality of indexed registers.



ABSTRACT

A processing device includes an optimizer to migrate objects from an external memory of a network processing device to registers connected to a processor. The optimizer further aligns and eliminates redundant initialization code of the objects.

$$(BASE + OFFSET) == (BASE | OFFSET)$$
Fig. 1**Fig. 2****Fig. 3**

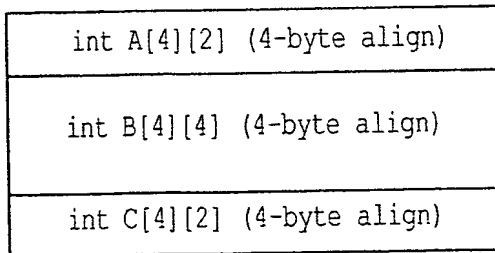


Original Data

FIG. 4A

- 1 Access Address A[i][0]
- 2 Access Address A[i][1]
- 3 Access Address B[i][0]
- 4 Access Address B[i][1]
- 5 Access Address B[i][2]
- 6 Access Address B[i][3]
- 7 Access Address A[i][0]
- 8 Access Address A[i][1]

Pseudo code sequence of accessing A, B, C

FIG. 4B

Data in local memory

FIG. 5A

Set the base address to A[i][0]
 Access Address A[i][0] (A[i][0]+0)
 Set the base address to A[i][1]
 Access Address A[i][1] (A[i][1]+0)
 Set the base address to B[i][0]
 Access Address B[i][0] (B[i][0]+0)
 Set the base address to B[i][1]
 Access Address B[i][1] (B[i][1]+0)
 Set the base address to B[i][2]
 Access Address B[i][2] (B[i][2]+0)
 Set the base address to B[i][3]
 Access Address B[i][3] (B[i][3]+0)
 Set the base address to C[i][0]
 Access Address C[i][0] (C[i][0]+0)
 Set the base address to C[i][1]
 Access Address C[i][1] (C[i][1]+0)

Pseudo code sequence of
 accessing A, B, C with
 initialization code of local
 memory based address inserted

FIG. 5B

3/7

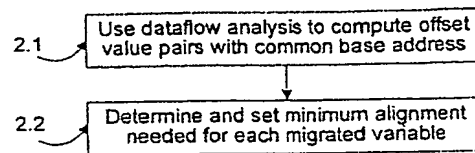


Fig. 6

```

Compute the GEN, KILL, IN, and OUT for each flow node, fill in the hash table {base address, set of offset value pair}
for (each base address in the hash table)
  VAR is the variable accessed by the base address
  for (each offset value pair for this base address)
    int CURR_VAR_ALIGN = current alignment of VAR
    int CURR_BASE_ALIGN = current alignment of the base address
    if (CURR_BASE_ALIGN does not satisfy the condition in Figure 1 for one of the offset value in the pair)
      int NEEDED_BASE_ALIGN = the minimum base address alignment needed to satisfy the condition in Figure 1 for
      all offset values in the pair
      int new_align = CURR_VAR_ALIGN * NEEDED_BASE_ALIGN / CURR_BASE_ALIGN
      if (new_align <= MAX_ALIGN(VAR))
        set the alignment of VAR to new_align
        if (the alignment change does not make the base address satisfy the condition in Figure 1 for all offset values in
        the pair)
          restore VAR's alignment to CURR_VAR_ALIGN
        end if
      end if
    end if
  end if
end if
  
```

Fig. 7

int A[4][2] (8-byte align)
int B[4][4] (16-byte align)
int C[4][2] (8-byte align)

Data with adjusted alignment

FIG. 8A

```

Set the base address to A[i][0]
Access Address A[i][0] (A[i][0]+0)
Set the base address to A[i][0]
Access Address A[i][1] (A[i][0]+4)
Set the base address to B[i][0]
Access Address B[i][0] (B[i][0]+0)
Set the base address to B[i][0]
Access Address B[i][1] (B[i][0]+4)
Set the base address to B[i][0]
Access Address B[i][2] (B[i][0]+8)
Set the base address to B[i][0]
Access Address B[i][3] (B[i][0]+12)
Set the base address to C[i][0]
Access Address C[i][0] (C[i][0]+0)
Set the base address to C[i][0]
Access Address C[i][1] (C[i][0]+4)

```

Pseudo code sequence of accessing A, B, C
after insert code to initialize the
local memory base address

FIG. 8B

int A[4][2] (8-byte align)
int B[4][4] (16-byte align)
int C[4][2] (8-byte align)

Data with adjusted alignment

FIG. 9A

```

Set the base address to A[i][0]
Access Address A[i][0] (A[i][0]+0)
Access Address A[i][1] (A[i][0]+4)
Set the base address to B[i][0]
Access Address B[i][0] (B[i][0]+0)
Access Address B[i][1] (B[i][0]+4)
Access Address B[i][2] (B[i][0]+8)
Access Address B[i][3] (B[i][0]+12)
Set the base address to C[i][0]
Access Address C[i][0] (C[i][0]+0)
Access Address C[i][1] (C[i][0]+4)

```

Pseudo code sequence of accessing A, B, C
after insert code to initialize the
local memory base address

FIG. 9B



5/7

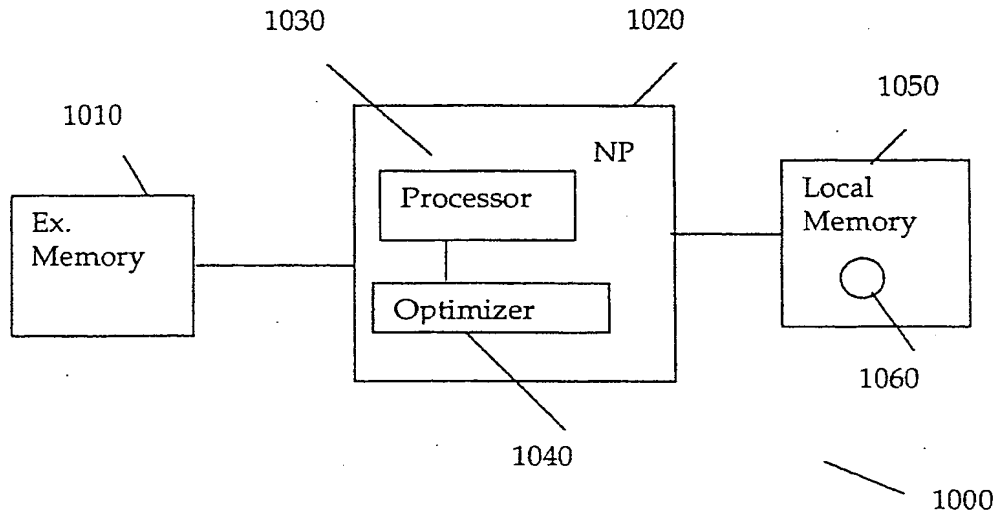


Fig. 10



6/7

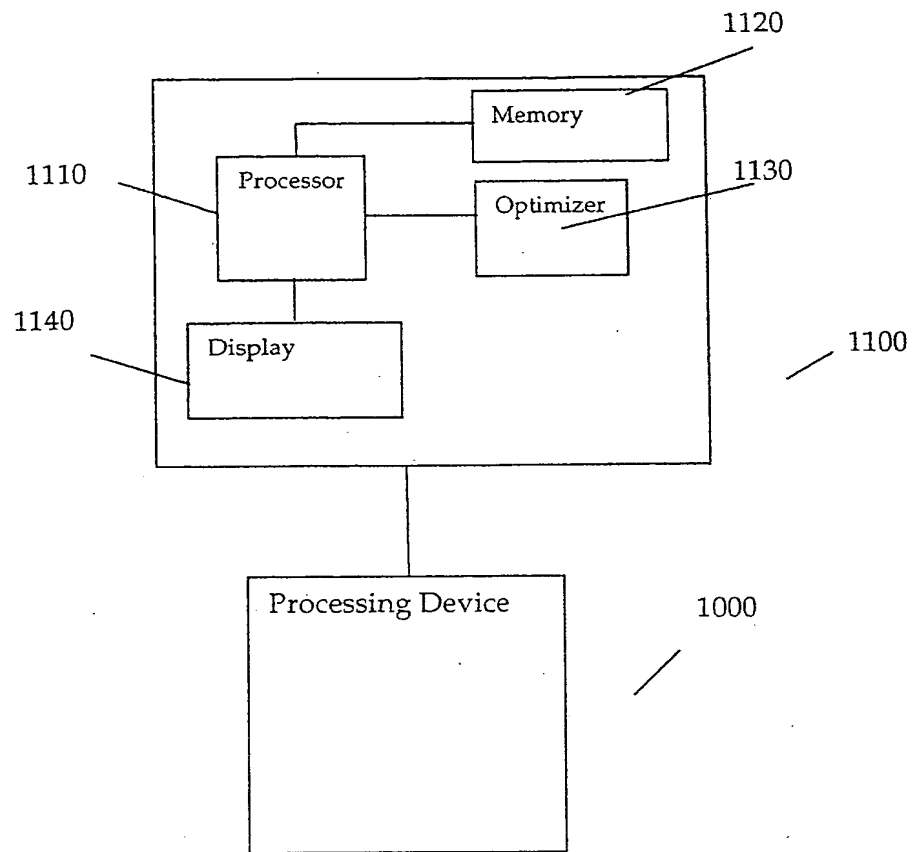


Fig. 11

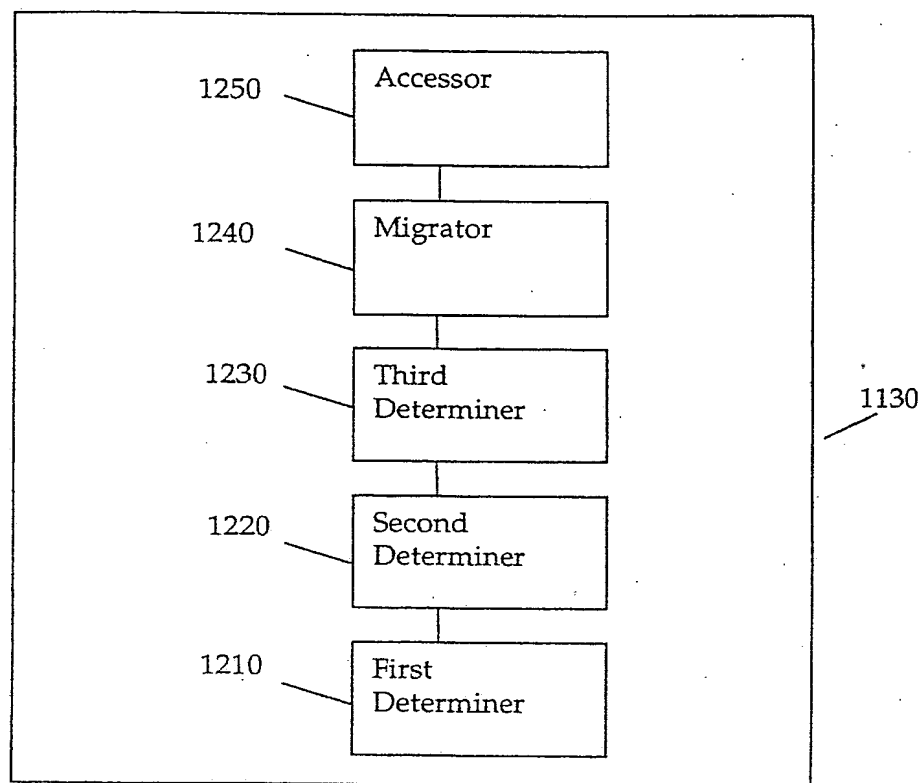


Fig. 12